A Mobile World of Security

Christine Neuberg, Panos Papadimitratos, Christina Fragouli and Ruediger Urbanke
EPFL, Switzerland
christine.neuberg@epfl.ch, panos.papadimitratos@epfl.ch, christina.fragouli@epfl.ch, rudiger.urbanke@epfl.ch

***Abstract*—Mobile users are increasing fast in numbers, new types of services and applications become available, and new mobile systems (e.g., for intelligent transportation) emerge. Meanwhile, the need for securing communication in such large scale, highly dynamic systems grows. But the organizational complexity and operational costs make traditional security solutions hard to deploy at the rate new mobile applications are rolled out. The challenge that lies ahead is how to provide versatile security compatible with the large deployed mobile networking infrastructure. We propose a novel approach to establish cryptographic keys. Our basic observation is that users are often very mobile and, as they interact with the infrastructure, each of them can leave a unique trace behind. Unlike many works that seek to identify structure in the mobility of a population, we leverage the inherent randomness of the mobility of individuals (and thus the randomness of mobile-infrastructure interactions) to establish shared secret keys between each mobile node and the infrastructure. With an underlying readily available source of uncertainty, such keys can be generated as needed, on the fly, to enhance the system and user security in many ways. We find that with no or little change to existing mobile communication systems, users can generate a common secret with the infrastructure at a rate of roughly 0.1 bits per second.**

I. Introduction

A core requirement for future wireless systems is to support communication for large numbers of mobile users, while still offering security and privacy. Nowadays, users are eager to use their smart phones for a multitude of services and applications: data access while commuting to work, social networking, urban sensing, and vehicular communications. Connectivity to fixed networking infrastructures gets more and more extensive, and interactions with other users become less rigid and more ad-hoc. As we move from traditional cellular telephony to more fluid and mobile settings, it becomes increasingly important and difficult to achieve security and privacy.

A common denominator in today's approaches for security and privacy is that mobility is treated as a hurdle to overcome, a challenge for the system's performance or even functionality. Consider for example public key infrastructures (PKIs), a long-known yet complex to organize solution: In large-scale dynamic systems, with multiple domains (as is the case for cellular and vehicular communication systems), special care must be given to the setup and the ability to authenticate any entity from any other domain. Thus far, deployed cellular data and voice systems have not adopted PKIs; rather, they rely on symmetric key authentication methods based on pre-established secrets (SIM cards) and cross-organizational verification of identity and authorization (e.g., for billing and accounting).

The need for alternative methods for key management, notably in the wireless domain, spurred a recent trend: to exploit the physical layer communication as a source of randomness for secret key generation. A number of schemes, surveyed in Sec. VI, leverage the wireless channel variability so that two nodes in range of each other utilize reciprocal observation of signals to establish a shared secret key.

Our work is close in spirit, even though we propose a fundamentally different approach: to leverage mobility and its inherent randomness as the basis for key generation. Our starting point is that users are often quite mobile and in most cases they interact with networking infrastructures; in doing so, each of them leaves behind a unique trace. Many works seek to identify structure in the mobility of a population, to anticipate user mobility and offer services. Instead, we exploit the inherent randomness of each individual mobility trace. For example, the sequence of base stations a driving user's smart phone connects to and the times these encounters occur constitute shared information between the user and the infrastructure.

It is this information exactly that allows establishing a shared secret, and also renew it over time, simply thanks to the user mobility. Having such a source of secret keys opens a new range of opportunities for users to address security and privacy needs, complementing existing techniques. In this paper, we investigate the feasibility of this novel approach and we shed light on what rates of secrecy can be achieved. Our main contributions are:

1) A new approach for secret key generation, applicable to a wide range of systems that involve mobile nodes and a wireless communication infrastructure. We investigate design choices and propose a backward compatible protocol with low overhead cost for existing cellular systems.

2) The performance evaluation of our scheme, through analytical modeling as well as simulations. We find that over a practically acceptable period of time a mobile node and an entity on the back-end of the infrastructure can derive common information, sufficient for establishing a strong secret shared key; most notably so, even in the presence of a strong adversary.

The rest of the paper is organized as follows. Sec. II describes the main idea behind our approach, and introduces basic definitions; Sec. III presents our protocol design and implementation; Sec. IV introduces our analytical model; Sec. V presents our evaluation; Sec. VI surveys related work and discusses the positioning of our work; and Sec. VII concludes the paper.

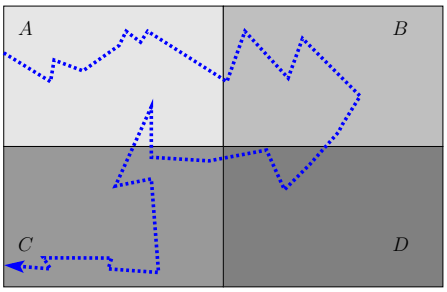

Figure 1. Bob's (the mobile's) itinerary in the area of Alice (the infrastructure), which operates the four base-stations, A,B,C, and D.

II. Main Idea and Setup

Currently, mobility is perceived primarily as a hurdle to be overcome in mobile computing systems that need to support secrecy and secure communication. Our position is that, on the contrary, mobility can be a source of secrecy and thus enable security enhancements. Moreover, such enhancements can be achieved using simple schemes, building on well-established existing infrastructure. We illustrate next our approach assuming a cellular infrastructure.

A. Main Idea

Consider a user, Bob, who enters a geographical area at some time t_0 and drives around until he exits the area at a later time $t_{\mathfrak{E}}$. His itinerary can be described by his geographical position at each time t within the interval of interest, $t_0 \leq t \leq t_{\mathfrak{E}}$. Let $l_x(t)$ and $l_y(t)$ be continuous random variables describing his (l_x, l_y) coordinates at time t; then, $L(t) = (l_x(t), l_y(t))$ is a continuous-time stochastic process that describes Bob's itinerary. Fig. 1 depicts a curve $L(t)$.

Our first observation is that such a curve has inherent randomness: even if this is the itinerary Bob uses on an every day basis, e.g., to commute to work, he may not use the exact same roads; even if he does use the same roads, he may not start at exactly the same time, as he may encounter different traffic conditions. Thus, he will not be at the same positions at the exact same times. Our second observation is that, currently, infrastructures for mobile communications already have some knowledge regarding this itinerary, or they could readily record such information in detail with little effort. This knowledge we can exploited to form the basis of a common secret, between Bob and Alice, the infrastructure.

Consider a cellular infrastructure and assume that our area is covered by four base-stations, A, B, C and D, as shown in Fig. 1. As Bob moves, his mobile device is at any time connected to one of these base stations,[1] depending on his location and the signal strength. This is information that both Bob and the cellular infrastructure have or can easily record. For example, Bob enters the cell of base-station A at time $t_0 = 0$, base-station B at $t_1 = 3$, basestation D at time $t_2 = 5$ returns to A at time $t_3 = 7$, then proceeds to base-station C at time $t_4 = 8$, etc. That is, we can in a sense[2] map $L(t)$ into another stochastic process $X(t)$, which describes the base station that Bob is connected to at time t. $X(t)$ offers a view of Bobs itinerary as perceived by the cellular infrastructure. Note that $X(t)$ takes values in the finite set $\mathcal{M} = \{A, B, C, D\}$, while the time variable t is continuous.

Our goal is to exploit the common knowledge of $X(t)$ between Alice (the infrastructure) and Bob (the mobile), in order to build a practical protocol. We thus first sample $X(t)$ every δ_s sec to create the discrete signal

$$X[i] \triangleq X(i\delta_s),$$

and then we keep n samples $X[i]$ in a vector

$$X \triangleq [X[0],\ X[1], \ldots X[n-1]]. \tag{1}$$

This vector is what our protocol uses as a shared secret between Bob and the infrastructure. For example, in Fig. 1, if starting at time $t_0 = 0$ we sample $X(t)$ every $\delta_s = 1$ sec and keep $n = 9$ values we get the vector $X_1 = [A\ A\ A\ B\ B\ D\ D\ A\ C]$ while if we sample every 2.5 sec and keep 4 values we get the vector $X_2 = [A\ A\ B\ A]$. Note how the sampling period δ_s affects the observed vectors X_1, and X_2; for example, one cannot tell that X_1 and X_2 come from sampling the same stochastic process $X(t)$. In the evaluation of our protocol, we will investigate its performance as a function of the sampling interval δ_s.

B. Adversary

The need for key establishment and secrecy implies the presence of an adversary, Eve, any entity that wishes to obtain the key shared by Alice and Bob. In our context, the adversary could attempt to learn the vector X, and from this derive the secret key. To do so, Eve would need to eavesdrop Alice and Bob exchanges over a long period of time, intercepting messages and recording the times Bob is connected with each base station, in order to establish X. In fact, Eve would have to achieve this in spite of fluctuations of the wireless links, impairments of the wireless communication between Bob and base stations, the uncertainty on the choice of base station (recall: Bob may choose among multiple base stations within range), and the time of the Alice-Bob key establishment.

We assume that Eve is a passive eavesdropper that can have presence in the area of Alice, i.e., covered by the infrastructure nodes. Her presence, e.g., through the deployment of own eavesdropping devices, is assumed bounded, that is, within a part of the entire area, however, she may be able to eavesdrop communication with multiple base stations. Assuming an arbitrary non-unique label assignment to all base stations, we consider Eve an adversary that can *intercept all* communication between Bob and *all* base stations with the same label. We emphasize that *all* means literally all messages

[1] Although in practice Bob may be connected to more than one base-stations, for example during soft hand-off, there exists always a single primary base station, the one responsible for the signaling.

[2] There is a degree of randomness in this mapping. Bob may at the same location connect to different base stations depending on the channel conditions and the traffic the base stations support; however, $X(t)$ itself contains a large amount of randomness, and this is what we want to exploit.

(in spite of practical wireless channel and other limitations), thus being a worst-case scenario for our scheme.

This is a particularly strong adversary. As it will become clear in Sec. IV and Sec. V, the base station labeling we consider is such that Eve can intercept all messages to **25%** of all base stations. Consider for example a medium size country and one major cellular telephony and data provider with more than 3000 base stations (in with 38 Location Areas and each base station with multiple cells): capturing all activity in more than 750 base stations (or more than 2000 cells) implies ample resources for the adversary.

C. Performance Metrics

In order to evaluate the quality of our key, we need to quantify the inherent randomness in the collected vectors X. If the basestation are assigned labels from a set $\mathcal{M}$ of size m, then we can consider this vector X to be a discrete random variable that takes m^n values.

A vector is considered to be truly random if it is not compressible; i.e., there is no scheme, which in average assigns a shorter description to the vector than the vector itself. This is the case if each element $X[i]$ of the vector is chosen uniformly at random from the set $\mathcal{M}$, and as a result, X takes the m^n values with equal probability. In this case, we say that the vector carries $n\log_2(m)$ bits of information (this is the amount of information we need to describe the vector in average). In general, the value of consecutive elements ($X[i]$ and $X[i+1]$) might be dependent, which reduces the uncertainty contained in the vector X. The incompressibility or inherent randomness of a vector can be characterized by its *entropy* [1].

Definition 1 (Entropy $\mathbb{H}(X)$) *Let X be a discrete random variable with distribution p. The entropy is defined by:*

$$\mathbb{H}(X) = -\sum_{x\in X} p(x)\log_2 p(x).$$

How does this entropy behave if we consider vectors of larger and larger size? Or otherwise put, how much entropy do we gain if we add one sample to an already long vector. This is characterized by the *entropy rate.* For a stationary process the entropy rate is a lower bound on the entropy of a finite vector, normalized by its length. This is intuitive since the entropy rate takes all dependencies into account, even those that go beyond the boundary of the vector. For very large vectors, the two become identical. The entropy rate is therefore the most fundamental measure.

Definition 2 (Entropy Rate $\mathcal{H}(X)$) *Consider a stochastic process that corresponds to a vector X of infinite length. The* entropy rate *of X is then defined as*

$$\mathcal{H}(X) = \lim_{n\to\infty}\frac{1}{n}\mathbb{H}(X[1], X[2], \ldots, X[n]).$$

This limit exists for a stationary process.

In the sense of the above limit, if we use vectors of length n, where n is large, $\mathbb{H}(X) \approx n\mathcal{H}(X)$. Since Bob and Alice know X, they can create $\mathbb{H}(X)$ truly random bits in common, using one of the standard methods in the literature (e.g., [2]). In other words, they can create a secret key of length $\mathbb{H}(X)$ bits. If we assume that we collect one element every δ_s seconds, and if we consider a long vector, then we can achieve a *secrecy rate* of $\mathfrak{R}_s = \frac{\mathcal{H}(X)}{\delta_s}$ bits per second.

To quantify how much information an eavesdropper Eve learns regarding X, assume that Eve collects vectors A that are correlated to X (if we view again vectors X and A as discrete variables, their relationship can be captured through their joint distribution). We can then use the *conditional entropy* to learn how much information is in X that we cannot learn by observing A.

Definition 3 (Conditionally Entropy $\mathbb{H}(X \mid A)$) *Let X and A be two discrete random variable with joint distribution p. The conditional entropy is then defined by:*

$$\mathbb{H}(X \mid A) = -\sum_{(x,a)\in(X,A)} p(x,a)\log_2 p(x \mid a).$$

D. Why a new scheme for key generation?

The nearly pervasive wireless communication infrastructures (cellular, WiFi, mesh networks) offer a first-class opportunity: they are already capable of collecting data on the connectivity of mobile devices, in fact, as part of their regular operation in order to support their users. At the same time, mobile devices can also easily determine which infrastructure node they are connected to. In other words, our vectors X can be collected without any special purpose infrastructure.

Equally important, unlike the vast majority of alternative solutions (Sec. VI), our scheme is independent of the wireless communication technology itself and it does not require any modification of the mobile or infrastructure transceivers.

Moreover, it does not depend on low-level, at the wireless physical layer, measurements, thus it can enable key establishment between the mobile user device and a server "behind" the wireless infrastructure; this would not be possible with methods that leverage the wireless channel properties.

Finally, mobility can be an incessant source of randomness and thus secrecy: users can get essentially "for free" a key established while on the move. Further, they can have a continuous accumulation of secret bits over time, and refresh older (and perhaps compromised) keys; along the lines of the idea to recover loss of secrecy with newly dynamically established secrets [3].

III. Our Scheme

Our scheme establishes a secret key between two entities:

(i) The *infrastructure-side* entity, S, any machine (e.g., server) that lies on the back side of the wireless infrastructure (i.e., the wire-line network). The wireless infrastructure consists of a set of *base stations*, $\{I\}$, distributed across a geographical area, connected with wire-line links and nodes, such as routers and switches. S can access the $\{I\}$ infrastructure.

(ii) A *mobile node*, V, that roams in the geographical area covered by $\{I\}$. V can connect to base stations in $\{I\}$

when in range, possibly making use of a semi-reliable data link protocol to mitigate communication errors. Over time, V connects to multiple base stations, one at a time.

Each base station is assigned a label from a set $\mathcal{M}$. The *base station labeling* is known in advance to both S and V. Multiple base stations may be assigned the same label.

Trace Collection: To generate a shared key, V and S need to *collect a trace* of data on the connectivity of V with base stations in $\{I\}$. This is initiated by the mobile, V, which asks for the bilateral data collection by sending an *Initiate* (V, T_I, T_E, δ_s) message to S. Upon receipt, S notifies V if it accepts or rejects this request. If accepted, logging of $V \leftrightarrow I$ connections starts at time T_I, with one sample every δ_s seconds, and ends at time T_E. Let $X(t)$ be the stochastic process describing the label of the base station that V is connected to at time t. The collected samples $X[i] = X(i\delta_s)$ are stored in a what we term the *trace* vector $X = [X[0],\ X[1], \dots X[n-1]]$, where $n = \frac{T}{\delta_s}$ and $T = T_E - T_I$ is the total sampling duration.

The samples are retained with appropriate logging at S and V, at each predefined time, t, and for each sample. V is in direct contact with one base station, and S retrieves the $V \leftrightarrow I$ connection information from the infrastructure (Note: this is either available at the side of the infrastructure or a low-cost probe can be triggered by S).

Trace Consolidation: V and S need to *consolidate* their traces, i.e., account for any errors in the trace creation, so that the key extraction is done on the same trace. Let X_V and X_S be the traces on the mobile's and the infrastructure's sides respectively. An error occurs, if these differ in one or more positions. To detect and correct errors, we establish a short $V \leftrightarrow S$ communication (shortly after time T_E). To reveal minimal information about X_V and X_S during consolidation, even in the presence of an adversary that perfectly intercepts the communication, well-known mechanisms for privacy amplification [4] can be used.

Key Extraction: Finally, S and V use the common information, X_S (which is identical to X_V) to derive a shared secret key $K_{V,S}$. Both parties implement a set of hash functions, agreed upon in advance from an appropriate class of hash functions (see universal-2 class [5] and [6], [7]). The output $f_H(X_S) = f_H(X_V)$ is XOR-ed with a predefined string, K_0 of the same length, to obtain

$$K_{S,V} = K_0 \mathtt{xor} f_H(X_S). \tag{2}$$

K_0 can be publicly known, and can be fixed (e.g., a function of the V and S identities).

The secret key extraction capitalizes on the randomness in the V mobility pattern to create random bits. Part of the mobility pattern is the amount of time V spends connected to a particular base station before switching. We term this the *sojourn* time $\mathcal{S}$, a random variable with a probability distribution that depends on the application, the system, and the user. The sojourn time distribution forms an *input to our scheme design*, used to assign values to its basic parameters.

Design Parameters: The trace collection is controlled by three basic parameters:

1) The *alphabet size*, i.e., the size of the set $\mathcal{M}$, or the number of distinct labels we assign to the base stations.
2) The *sampling interval* δ_s, i.e., how often we sample.
3) The *sampling duration* $T = T_E - T_I$, i.e., for how long we sample.

Parameter choice guidelines: (1) We propose to employ $\mathcal{M}$ of a size equal to the maximum number of neighbors a base station can have; and a labeling that assigns a different label to each neighbor. This stemmed from our analysis, in Sec. V, as such alphabets yield the same amount of randomness with larger ones (e.g., with a distinct label for each base station), while allowing reduced storage and processing complexity. Knowledge of the infrastructure deployment statistics suffice for that purpose. (2) We suggest to select a sampling interval of the same order of magnitude as the average sojourn time. Without considering wireless radio peculiarities, this would capture the mobility induced randomness. Coarse knowledge on the mobility patterns of users or even better statistics on the connectivity to $\{I\}$ can give guidelines for this. (3) Finally, the sampling duration T can be selected sufficiently large, to allow to extract the required number of random bits.

Other practical considerations: The trace collection requires that S and V have their *clocks synchronized*. S and all base stations in $\{I\}$ can be synchronized with the help of existing protocols (e.g., the Internet standard Network Time Protocol (NTP), and V can be synchronized with the base station it connects to. Synchronization cannot be perfect, however, as clock errors are in principle much smaller than the sampling intervals we consider; thus, the resultant small amount of potential errors can be overcome with the help of trace consolidation.

Example of Infrastructure Compatible Deployment: The information we require is simple to obtain in any wireless communication infrastructure, as they all uniquely identify their infrastructure nodes and they can probe them, and through them the connectivity of mobile nodes. A characteristic example is that of cellular voice and data networks, e.g., GSM/GPRS [8], [9] and UMTS [10]. As they are a widely deployed, essentially pervasive, we discuss our scheme in their context.

At any point in time, the mobile can determine the identity of the cell it is connected to. Recall that each base station (BTS in GSM terminology) has typically 3 cells. Thus, our X_S vector will be composed of cell identities (Cell IDs). Each Cell ID is periodically broadcasted, thus the mobile can easily identify all cells in range and elect the one it considers as a connection. Existing operating systems for mobile devices (e.g., Android) provide the appropriate interface for the mobile to obtain easily this information.

On the other hand, the V connectivity is maintained only when the mobile initiates or receives a call or sends/receives data. While this case and in order to maintain the end-to-end connectivity of the mobile, the infrastructure keeps closely track of all changing connections of the mobile. Otherwise, to reduce signaling overhead when the mobile is idle, the

'location' (i.e., the cell ID where the mobile is connected to) is updated only when the mobile crosses the boundary of the so-called location areas. As those areas are typically large (depending various parameters, but typically from 30 to 400 cells per area), having only such infrequent updates would yield little information for our purpose.

We can resort to two alternative simple solutions to have up-to-date information for the trace collection of our scheme. First, the mobile, V, can transmit a short message (e.g., a data packet or SMS to a predefined "sink" node) shortly before the time of the i-th trace sample. This way, V ensures that the infrastructure has available the appropriate connection data sample. All that S would need to do is to probe the Mobile Switching Center (MSC) and/or the Visitor Location Register (VLR) to obtain the cell ID (and convert it to its label through our mapping). Another approach would be to have the S initiate a probe shortly before the time of the i-th trace sample. This is possible with little cost, for example, by using the method of [11], which sends a special-purpose void SMS message to the mobile V; this does not disturb the user but still triggers an update of its "location", i.e., the Cell ID, at the side of the infrastructure. Finally, note that the openness of cellular infrastructures to such extraction of information by third parties is already a fact, e.g., for tracking of cell phones by specialized services, through the opening of specific interfaces, or for provision of location based services.

IV. Modeling and Analysis

Let us now introduce a model that allows us to analyze the performance of a given system, i.e., the quality of the produced key. The performance metrics we use are entropy and entropy rate (see Sec. II). Our model takes as input the sojourn time distribution and calculates the entropy and entropy rate of the resulting traces. More precisely, we model the collected vectors as being produced by a *renewal* process, where the inter arrival times of the renewal process are equal to the sojourn times. We consider both the cases where an adversary is present or not.

We validate our model in Sec. V, using as input the sojourn time distribution we collect from simulated traces. More precisely, we compare in Sec. V the analytic performance predictions with the results form actual simulations. We find very good agreement.

A. Model

Recall that $X(t)$ is the stochastic process describing the labels of the base station that user Bob is connected to at time t, and X is the trace of length n

$$X = [X[0],\ X[1], \dots X[n-1]], \tag{3}$$

where $X[i] = X(i\delta_s)$ and δ_s is the sampling interval. The randomness of the trace X is due to the randomness in the routes and also due to the random amount of time users are connected to various base stations. The process $X(t)$ is quite complicated, and so is X. Let us therefore introduce a model which is analytically more tractable but preserves the key characteristics of $X(t)$.

Also recall that the *sojourn* time S is the random variable that describes the amount of time that a user spends connected to a particular base station before switching. For a fixed sampling period δ_s, we measure the sojourn time in multiples of δ_s, i.e., $S \in \mathbb{N}\,\delta_s$. E.g., $S = 3\delta_s$ means that for 3 consecutive sampling times a user is connected to a particular base station, but that at the 4-th sampling time instance he has switched. Our model takes as input the distribution of S for a given parameter δ_s; this can be calculated either through simulations, or through modeling.

Definition 4 (Interarrival Times U) *As mentioned above, S takes values in $\mathbb{N}\,\delta_s$. Associate to S the integer-valued random variable U, where $p_i = P(U = i) = P(S = i\delta_s),\ i \geq 1$. We call U the* inter-arrival *time.*

We now postulate that we can model the connection process by a renewal process with inter-arrival times distributed according to U. With respect to the labels we assume that for every new connection the label is chosen uniformly at random from the label set $\mathcal{M}$. This is a reasonable model if we assume that we choose a relatively small label set. Why would we be content with using a small label set? The analysis shows that a significant source of randomness is contained in the "timing" information. Further, one would expect that even if we assigned unique labels to each base station the amount of randomness does not significantly increase. This is true since, given the identity of a base station, we know its location, and for reasonably sampling intervals δ_s, the "next" base station will very likely be one of its nearest neighbors. This number of neighbors is typically small (perhaps 4) and it is this number and not the absolute number of base stations which limits the entropy rate. Given these considerations we will stick to small label sets. This might be beneficial also in terms of storage and processing.

Definition 5 (Renewal Process R and Label Process Y) *Let R denote a renewal process with inter-arrival times distributed like U. More precisely, let $F_U(x)$ denote the distribution function corresponding to U. Let $G_U(x) = \frac{1}{\mu_p}\int_{z=0}^{x}(1 - F_U(z))dz$. Let U_1 be distributed according to G_U and let $U_2, U_3, \dots$ denote a sequence of independent random variables, distributed according to F_U. Let $R_0 = 0$ and $R_n = \sum_{i=1}^{n} U_i,\ n \geq 1$.*

Let $Y[i]$ denote our analytical model for $X[i]$. We pick $Y[R_n],\ n \geq 0$, uniformly at random from $\mathcal{M}$. For $i \in [R_n + 1, \dots, R_{n+1} - 1]$, we define $Y[i] = Y[R_n]$. Finally, define Y as $Y \triangleq [Y[0],\ Y[1], \dots Y[n-1]]$. As for X we call Y the trace.

Discussion: The reason we have chosen the distribution of U_1 in this particular way (and different from the distribution of all other U_i) is that this choice makes the process stationary. In particular, for $i \geq 0$ the process as defined behaves as if the

renewal process had started in the infinite past. This simplifies our analysis.

Definition 6 (Adversary Process) *Let the trace for the user be as in (3). We assume that the (*strong*) adversary is present in all base stations having a specific label $\mathfrak{m}_i \in \mathcal{M}$, i.e., the adversary is present in a fraction $\frac{1}{m}$ of all base stations. The process A for the strong adversary is then defined as*

$$A[j] = \begin{cases} 0, & \text{if } Y[j] \neq \mathfrak{m}_i, \\ 1, & \text{if } Y[j] = \mathfrak{m}_i. \end{cases}$$

The stochastic process of the strong adversary is defined as $A \triangleq [A[0],\ A[1], \dots A[n-1]]$. The process A is stationary because it is fully determined by the stationary process Y.

B. Entropy and Entropy Rate of Y

Lemma 1 *Let Y be the label process as given in Def. 5. For $n \geq 1$, $\lim_{\delta_s \to 0} \mathbb{H}(Y) = -\sum_{i=1}^{m} \frac{1}{m} \log_2 m = \log_2 m$ and $\lim_{\delta_s \to \infty} \mathbb{H}(Y) = -\sum_{i=1}^{m^n} \frac{1}{m^n} \log_2 \frac{1}{m^n} = n \log_2 m$. For $n = 1$, $\mathbb{H}(Y) = \log_2 m$. For $n = 2$, $\mathbb{H}(Y) = \log_2(m) - \frac{(\mu_p-1)m+1}{\mu_p m} \log_2(\frac{(\mu_p-1)m+1}{\mu_p m}) - \frac{m-1}{\mu_p m} \log_2(\frac{1}{\mu_p m})$. For $n = 3$,*

$$\begin{aligned} \mathbb{H}(Y) =& m\ \pi_{xxx} \log_2 \pi_{xxx} + 2m(m-1)\ \pi_{xxy} \log_2 \pi_{xxy} \\ &+ (m(m-1)(m-2) + m(m-1))\ \pi_{xyz} \log_2 \pi_{xyz} \end{aligned}$$

with

$$\pi_{xxx} = \frac{1}{m} \frac{\mu_p - 2 + p_1}{\mu_p} + \frac{2}{m^2} \frac{(1-p_1)}{\mu_p} + \frac{1}{m^3} \frac{p_1}{\mu_p},$$
$$\pi_{xyy} = \frac{1}{m^2} \frac{(1-p_1)}{\mu_p} + \frac{1}{m^3} \frac{p_1}{\mu_p}, \quad \pi_{xyz} = \frac{1}{m^3} \frac{p_1}{\mu_p}.$$

Finally, the entropy rate $\mathcal{H}(Y)$ is given by

$$\mathcal{H}(Y) = \frac{\mathbb{H}(\{q_i\}) + \log_2(m-1)}{\mu_q},$$

where q and μ_q are defined as follows. Define $p(x) = \sum_{i=1}^{\infty} p_i x^i$. Let $q(x) = p(x) \frac{m-1}{m-p(x)}$. Let $\{q_i\}$ denote the corresponding probabilities, i.e., develop $q(x)$ as a Taylor series around $x = 0$. Let $\mathbb{H}(\{q_i\})$ be the entropy of $\{q_i\}$ as stated in Def. 1 and let $\mu_q = \sum i q_i$.

Proof: Due to space constraints let us only give the proof for the entropy rate since this is the most fundamental quantity. We compute the entropy rate via the expression $\lim_{n\to\infty} \frac{1}{n} \mathbb{H}(Y[1], Y[2], \dots, Y[n])$. Consider a large block length n. Our label process has two degrees of freedom. First, there is the degree of freedom of how the block length n is partitioned into smaller segments. This comes from the connectivity pattern. Second, for each small segment of the partitioning we have the degree of freedom of how to label this segment with an element from $\mathcal{M}$. These two degrees of freedom are however not independent. When a user switches a base station she picks a new label. But it might happen that the new label is identical to the old one. Therefore, from the process Y we cannot in general identify the times a user switches base stations.

Let us therefore consider an equivalent process in which these two degrees of freedom are cleanly separated. Any time a user switches a base station she has a $1/m$ chance of picking an identical label. Let us therefore redefine the inter-arrival times by joining those segments that have the same label together. Recall that$\{p_i\}$ denotes the probabilities of the inter-arrival times U_i. Define $p(x) = \sum_{i=1}^{\infty} p_i x^i$. Let $q(x) = p(x) \frac{m-1}{m-p(x)}$. Let $\{q_i\}$ denote corresponding probabilities, i.e. develop $q(x)$ as a Taylor series around 0.

Consider now a new label process that has inter-arrival times distributed according to $\{q_i\}$. For this process, every time a user switches, the new label is chosen uniformly at random from the set $\mathcal{M}$ *excluding* the old label. In other words, the label is chosen uniformly at random from a set of size $m-1$.

Assume now that we pick k segments according to $\{q_i\}$, where k is very large. Roughly kq_i of those segments will have length i, so the total length will be close to $\sum_{i\geq 1} q_i i = \mu_q$. There are approximately $2^{k\mathbb{H}(\{q_i\})}$ such arrangements and these arrangements are roughly equally probable. Further, for each of these arrangement we can still decide on the labels which adds $(m-1)^k$ degrees of freedom. The total number of (roughly equally likely) arrangements is therefore

$$2^{k\mathbb{H}(\{q_i\})}(m-1)^k = 2^{k[\mathbb{H}(\{q_i\}) + \log_2(m-1)]}.$$

Taking the logarithm, and dividing by the expected length $k\mu_q$ in order to get the entropy per symbol, gives the stated result. ∎

Discussion. We have only given entropy expressions for $n = 1$, 2, and 3. Although it is possible to derive expressions for larger n as well, as discussed in Sec. II, the operationally most significant quantity is the entropy rate. In the following we examine how this quantity behaves as a function of the systems parameters.

For either small or large values of δ_s it is easy to give accurate and simple expressions for the entropy rate. Consider first not too large values of δ_s. In this case the entropy rate is dominated by the randomness inherent in the "timing", i.e., by the term $\mathbb{H}(\{q_i\})$ and we can ignore the entropy contained in the labels, i.e., the term $\log_2(m)$. Further, the term $\mu_q \delta_s$ is roughly constant since it represents the average amount of time (in seconds) that it takes to switch labels and this time only depends in second order on the sampling interval (i.e., on δ_s). Finally, for small values of δ_s, $\mathbb{H}(\{q_i\})$ is increased by 1 bit every time we half the value of δ_s. If follows that the entropy rate (per second) has in this regime the expression $a - b \log_2(\delta_s)$. E.g., for the Lausanne scenario, that we will introduce in Sec. V, we have $a \approx 0.12$ and $b \approx 1/70$. In particular this means that the entropy rate tends to infinity if we let δ_s tend to 0. But of course, small δ_s comes at the cost of a large overhead; moreover, in every system the value of δ_s is lower bounded by the inherent timing accuracy which we can achieve.

For very large values of δ_s the entropy is eventually dominated by the term $\log(m)$. Further, for large values of δ_s, μ_q converges to 1. Therefore, in this regime the secrecy rate scales like $\log(m)/\delta_s$.

Scenario	Lausanne	Los Angeles
Dimensions	$6 * 6km$	$35 * 35km$
Distance between streets	$0.3km$	$1km$
Number of basestations	100	9520
Max. speed	$17m/s$	$34m/s$
Number of cars per simulation	$480 * 30$	$840 * 30$
Entry rate	$20 * 4$cars/20sec	$35 * 4/10sec$

Table I
SIMULATION PARAMETERS

Note finally, that if we scale the size of all cells by a factor α (or alternative, speed up cars by a factor $1/\alpha$), and at the same time scale δ_s by α as well, then the entropy rate scales by a factor $1/\alpha$. So small cell sizes increase the entropy as one would expect.

C. Entropy and Entropy Rate of Y under Adversarial Model

Lemma 2 *Assume that A is a renewal process as stated in Def. 6. The amount of information a user can hide from the adversary is characterized by*

$$\mathbb{H}(Y|A) = \mathbb{H}(Y, A) - \mathbb{H}(A) = \mathbb{H}(Y) - \mathbb{H}(A). \quad (4)$$

Recall that the adversary can control a fraction $\frac{1}{m}$ of all basestations. Define $h_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$. For all vector length $n \in \mathbb{N}$, $\lim_{\delta_s \to 0} \mathbb{H}(A) = h_2(\frac{1}{m})$. and $\lim_{\delta_s \to \infty} \mathbb{H}(A) = -\sum_{i=1}^{n} \binom{n}{i} \left(\frac{1}{m}\right)^i \left(\frac{m-1}{m}\right)^{n-i} \log_2 \left(\frac{1}{m}\right)^i \left(\frac{m-1}{m}\right)^{n-i}$. For $n = 1$, $\mathbb{H}(A) = h_2(\frac{1}{m})$. For $n = 2$, $\mathbb{H}(A) = h_2(\frac{1}{m}) + h_2(\frac{1}{m\mu_p}) + h_2(\frac{m-1}{m\mu_p})$. Finally, the entropy rate $\mathcal{H}(A)$ is given by

$$\mathcal{H}(A) = \frac{\mathbb{H}(\{q_i\}) + \mathbb{H}(\{r_i\})}{\mu_q + \mu_r},$$

where q and μ_q are given in Lemma 1 and where r and μ_r are defined as follows. Let $r(x) = \frac{p(x)}{m-(m-1)p(x)}$. Let $\{r_i\}$ denote the corresponding probabilities, i.e., develop $r(x)$ as a Taylor series around $x = 0$. Let $\mathbb{H}(\{r_i\})$ be the entropy of $\{r_i\}$ as stated in Def. 1 and let $\mu_r = \sum i r_i$.

Due to space constraints we skip the proof of the above lemma.

We note that for small to moderate values of δ_s the entropy rate of A also has the form $a - b\log_2(\delta_s)$ as was the case for the entropy rate of Y.

V. EVALUATION

A. Methodology

Testing environment: We test the performance of our protocol for the case of mobile users in cars. We produce data close to natural user behavior by using the SUMO traffic simulator [12]. SUMO generates routes that mimic the expected user behavior by using an algorithm called dynamic user assignment. It reflects the idea that traveling involves some time, cost or disutility that users would prefer to avoid. In essence, this algorithm finds the quickest route instead of the shortest path. We consider two types of underlying topology and traffic density:

1) the *Lausanne scenario* representing traffic in a small town, and
2) the *Los Angeles scenario*, representing traffic in a metropolitan area with a network of highways.

In place of maps, we use simplified square grids networks with parameters corresponding to those of the two cities. All streets have the same length, the same number of lanes per direction and the same speed limit. The crossings are set according to the right-before-left-rule without traffic lights. The simulated area is covered by square cells of the same size, each cell served by a basestation. In the simulations, each basestation is randomly assigned one of four different labels; thus each label is assigned to roughly 25% of the base stations. Each car enters and exits this area at a randomly chosen border point. The precise parameters for the two scenario are listed in Table I.

Adversary: An adversary present in the system might be able to overhear some information as explained in Sec. II-B, and thus reduce the amount of collected random bits that can be used for key generation. Our main model consists of a *strong* adversary who can overhear all parts of the trace corresponding to a specific label; in the simulations we also consider a weak adversary who only knows the positions in the trace where the user is connected to one specific basestation.

Validation of theoretical model : To validate our modeling and analysis in Sec. IV, we empirically calculate the distribution of the sojourn times using our simulations, and use this distribution as input for our theoretical model. We then compare the theoretically derived performance from the model with that of the simulated system.

Performance metrics: Our performance metric is the *secrecy rate* R_s, that quantifies the number of generated random bits per second. In the presence of an adversary, this quantifies the number of random bits the adversary has no information about. For traces of length n, R_s is calculated as $\frac{\mathbb{H}(X)}{n\delta_s}$, where $\mathbb{H}(X)$ is computed from the simulations as described in the following. For the theoretical results, we use $R_s = \frac{\mathbb{H}(Y)}{n\delta_s}$, where $\mathbb{H}(Y)$ is calculated using our modeling. As n increases, this quantity approaches $\frac{\mathcal{H}(Y)}{\delta_s}$. Similarly, in the presence of the adversary we use $\frac{\mathbb{H}(X|A)}{n\delta_s}$ and $\frac{\mathcal{H}(Y|A)}{\delta_s}$.

Entropy computation and a practical challenge: For a fixed trace length n the entropy can be computed based on simulations in the following way. Lets say, we have generated a collection of vectors X from traces of all users in the system. This in turn allows us to estimate the probability distribution on the set of possible outcome vectors. We can then compute the entropy associated to this probability distribution.

In practice, this approach quickly reaches its limits. We need to calculate the distribution on the set $\mathcal{M}^n$, which has cardinality m^n. A realistic assumption would be to have approximately 100 micro cells in an 6km $*$6km large urban street area; if a car spends approximately 5mins in this area, and we use $\delta_t = 20$sec, the trace length would be approximately $n = 15$. Even if we assume that we only use $m = 4$ labels, i.e., we set $m = 4$ in order to reduce complexity, we still have to estimate

a probability distribution on a set of size $4^{15} = 2^{30}$, which is roughly a billion. This quickly becomes close to infeasible or at least impractical as m and n increase, and limits the scenario we can examine through simulations. This is where the theoretical modeling can help.

B. Results

Figure $2(a)$ shows the performance of our protocol as function of the sampling interval δ_s for the case of the Lausanne-scenario. We plot both the results obtained through simulations (dotted curves) as well as the results obtained from the theoretical model (continuous curves). Our simulations provided results for trace lengths $n = 2, n = 3, n = 4, n = 6$ and $n = 8$; we see that these results fit quite well to the curves derived from the theoretical model. We thus proceed to use the theoretical model to derive the upper and lower bound curves also depicted in the figure, which correspond to vector lengths $n = 1$ and $n \to \infty$, respectively. The latter is derived using the entropy rate, as we discussed earlier.

We see that as the trace length n grows, the curves converge towards the entropy rate (which, as we discussed earlier, gives a lower bound for any length): in fact we closely approach this curve even for relatively short length, e.g., $n = 8$. Curves for shorter length seem to deliver higher secrecy rates; however, this is misleading since these computation only apply if we collect a set of such vectors which are well separated but loose their meaning if we sample consecutive vectors due to the dependence of such vectors. Thus, the entropy rate gives a more realistic estimate of the performance we can expect to achieve. For example, in the Lausanne scenario, we expect to collect 128 random bits in approx. 25 minutes, if we sample three times per minute, while in the Los Angeles scenario we expect to collect 128 in approx. 15 minutes, if we double the sampling rate.

The remaining plots in Fig. 2 show the effect that the presence of a strong or a weak adversary has in the achievable secrecy rate. Note that the weak adversary, which is a very realistic case (an adversary overhearing all communications in a single basestation) has a negligible effect; the strong adversary, on the other hand, reduces the secrecy rate by approx. 40%. Indeed, such an adversary, by overhearing all communications in 25% of the basestations can deduce also much about about the conversation in between. To consider an extreme case consider the example of $m = 2$. In this case the strong adversary, by observing have the positions can in fact reconstruct the whole trace.

VI. Related Work and Discussion

A. Related Work

Limitations of traditional key management, notably those based on public key cryptography, spurred work that sought alternative methods to establish symmetric shared keys. The motivation has been to avoid trusted third parties and public key cryptography (e.g., Diffie-Hellman secret key establishment [13] that relies on the computational hardness of the discrete logarithm problem. Information-theoretic schemes for key generation based on correlated information have been the basis: simply put, two legitimate parties observe a source the adversary cannot even if it can intercept messages they exchange [14], [15]. The adversary can be a passive eavesdropper (as is also the case for the DH protocol), and the possibility of key establishment in the presence of active adversaries is proven [16], [17], [2].

Advances mobile computing were a catalyst, as small-footprint wireless platforms need to set up keys often with peers in an ad hoc manner. A range of schemes were proposed, leveraging the wireless channel properties to establish common information an eavesdropper is highly unlikely to obtain. Its reciprocity (radio wave propagation effects are the same in both wireless link directions) and the time (the channel changes over time) and spatial (it is uncorrelated for any two receivers more than few wavelengths apart) variability of the wireless channel make this possible: Alice and Bob can make channel measurements that Eve is practically unable to obtain. The early [18] proposed transmission of tones across an urban UHF channel, and other more recent methods followed.

Relying on Received Signal Strength (RSS) measurements, special care is needed to address the asymmetries of this statistic (the RSS is not the same in both directions). A three-way message exchange and sampling of the signal envelope was proposed in [19]; extracting randomness from deep fades in [20], along with information reconciliation techniques to securely correct inconsistencies; a level-crossing algorithm in [21], to trace channel impulse response estimates within an index, and an 802.11 implementation-based evaluation; also experimental implementation-based evaluation of various secrecy extraction techniques for 802.11 and an environment-adaptive improvements in [22]; a solution geared towards reduced energy consumption in wireless sensor networks (WSNs) and IEEE 802.15.4 compliant radios in [23]. [24] relies on the ultra-wide band (UWB) pulse response to derive an approximation and an upper bound of the mutual information for a general channel model; [25] relies on the UWB channel delay profiles. Observing that low channel variability can reduce the extracted information, special-purpose hardware, variable-directional antennas, and beam-forming techniques were proposed in [26], [27]. Phase reciprocity [28], and differential phase between tones [29] were also proposed.

Works leveraging the physical layer may require special hardware or modification of the wireless transceivers, at least in their ability to provide specific measurements. More important, even though they can offer significant secrecy rates (e.g., 10 bits/s by [30]), they are limited in their pair-wise and local operation (relying mostly on short-range radios). Without any transceiver modification, two devices could extract common information by track of the 'one time frames,' that is, wireless transmissions received at the first attempt (no retransmission) [3]; an eavesdropping adversary would need to eavesdrop all wireless transmissions for an extended period of time without errors.

But none of the above allows two remote devices that are not connected across the wireless medium to establish a shared

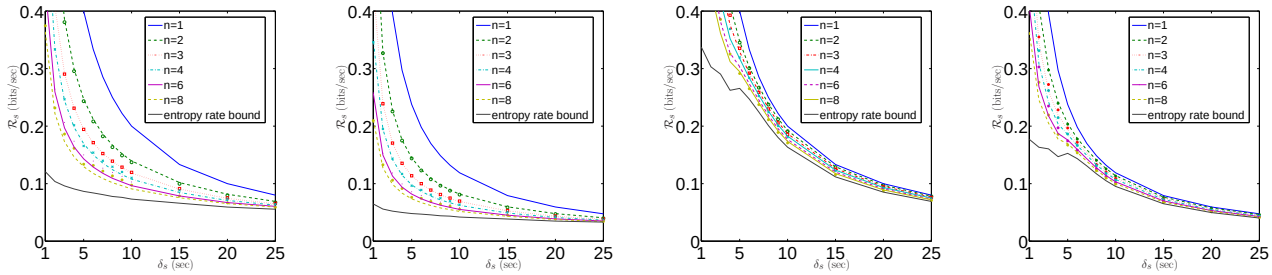

Figure 2. Lausanne scenario (a) without adversary and (b) with strong adversary, Los Angeles scenario (c) without adversary and (d) with strong adversary (for parameters see Tab. I).

key. This would be the case for a broad gamut of mobile applications beyond than what ad hoc 802.11, WSN, or UWB links can support. Our work seeks to close this gap. We propose an alternative mechanism independent of the wireless communication specifics, leveraging node mobility and the traces of their connectivity to wireless infrastructure; a scheme deployable on top of existing systems with no modification of transceivers.

B. Discussion

Our base position in this paper is that mobility has inherent randomness that can be exploited to establish common random bits at low cost. This observation is not constrained to a specific application or system. We exploited the interactions of the cellular infrastructure with users driving in a geopgraphical area; we could instead have considered pedestrian users connecting to femtocells, or even Internet users browsing webpages managed by a common server. We proposed a protocol that builds on the times of encounters with basestations; we could instead capitulate on encounters with other users as recorded by a satellite, speed at particular instances as measured by traffic infrastucture or any other mobility triggered random event witnessed by two independent entitities. We utilized the random bits for secret key generation; we could instead have addressed privacy considerations. For example, the common random bits could be used to establish a type of unique identity for a user, which cannot be taken over by any other; this would allow the user to reap the benefits of a stable, recurring interaction when needed, while, at the same time, not revealing her long-term identity.

The particular scheme we selected to examine is we believe very promising, not because it yields very high data rates, but because it operates at very low cost. Indeed, the mobility scenaria we investigated establish 128 bit secret keys in 15-25 minutes, which might at first seem long; however, this is well below the average daily commute time for many users, while the overall operation requires collection of $3-6$ samples per minute, a very low overhead for the capabilities of current mobile devices, that need not disrupt their main tasks. These almost "free" random bits can be used to enhance and complement other security systems, when needed.

Interestingly, we also find this simple scheme robust to adversarial attacks. Systems such as cellular networks or Wi-Fi infrastructures are well protected and centrally managed, with any intrusion or malfunction detected by their administrator. A passive eavesdropper can compromise an I node by 'tapping on' such a node, possibly indefinitely long, or by establishing an adversarial receiver in the vicinity of the I node, and ensuring that all communications between any V and that I node are received by the adversary. Clearly, both are hard to achieve for the adversary,[3] even for a single I node; nonetheless our scheme is operational even with a good fraction of them compromised. Moreover, a natural constraint stems from the inability of the adversary to be physically present throughout the system area and monitor (or compromise) all I nodes. If it could do so, it could deploy its own infrastructure or it could override all defenses of the infrastructure. An active eavesdropper, i.e., an adversary that injects messages, is orthogonal to our investigation, as we are not making any assumptions on prior trust and security associations between the mobile nodes and the infrastructure. Clearly, injection of arbitrary messages, for example, by a compromised I node or by impersonation of an I node could result in a failure of our protocol, i.e., the establishment of a shared key, $K_{S,V}$, between the mobile and the infrastructure. This cannot benefit the adversary: he could instead simply jam communications and erase messages.

VII. Conclusions

We proposed a protocol that leverages mobility for secret key generation, and evaluated its performance through modeling and simulation results. Although simple, our scheme can be implemented with low overhead, and is robust to adversaries.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 2006.

[2] U. Maurer and S. Wolf, "Secret-key agreement over unauthenticated public channels: Privacy amplification," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 839–851, April 2003.

[3] S. Xiao, W. Gong, and D. Towsley, "Secure Wireless Communication with Dynamic Secrets," in *IEEE INFOCOM*, 2010.

[3]Compromising well-managed infrastructure is not easy, while receiving all transmissions within a range, in a multi-access, interference limited environment, is not under the control of the receiver.

[4] C. Bennett, G. Brassard, and U. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, pp. 1915–1923, 1995.

[5] J. Carter and M. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, no. 18, pp. 396–407, 1997.

[6] S. Halevi and H. Krawczyk, "Strengthening digital signatures via randomized hashing," in *Advances in Cryptology, CRYPTO 2006, LNCS 4117*. Springer, 2005, pp. 41–59.

[7] "National Institute of Standards and Technology," www.nist.gov.

[8] "GSM world association," www.gsmworld.com.

[9] "ETSI GPRS," www.etsi.org.

[10] "UMTS," www.umtsworld.com.

[11] M. Ficek, T. Pop, P. Vláčil, K. K. Dufková, L. Kencl, and M. Tomek, "Performance study of active tracking in a cellular network using a modular signaling platform," in *ACM MobiSys*, Jun. 2010.

[12] D. Krajzewicz, M. Bonert, and P. Wagner, "The Open Source Traffic Simulation Package - SUMO," 2006.

[13] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[14] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography. i. secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, Jul. 1993.

[15] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[16] U. Maurer and S. Wolf, "Secret-key agreement over unauthenticated public channels: Definitions and a completeness result," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 822–831, April 2003.

[17] ——, "Secret-key agreement over unauthenticated public channels: The simulatability condition," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 832–838, April 2003.

[18] J. Hershey, A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Transactions on Communications*, vol. 43, no. 1, pp. 3–6, Jan. 1995.

[19] M. Tope and J. McEachen, "Unconditionally secure communications over fading channels," in *IEEE MILCOM*, vol. 1, 2001, pp. 54–58.

[20] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *ACM CCS*, Alexandria, Virginia, USA, 2007.

[21] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel," in *ACM MobiCom*, San Francisco, CA, USA, Sept. 2008.

[22] S. Jana, S. Premnath, M. Clark, S. Kasera, N. Patwari, and S. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *ACM MobiCom*, Beijing, China, Sept. 2009.

[23] J. Croft, N. Patwari, and S. Kasera, "Robust Uncorrelated Bit Extraction Methodologies for Wireless Sensors," in *ACM IPSN*, Stockholm, Sweden, Apr. 2010.

[24] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 364 –375, Sept. 2007.

[25] A. Kitaura, T. Sumi, K. Tachibana, H. Iwai, and H. Sasaoka, "A scheme of private key agreement based on delay profiles in uwb systems," in *IEEE Sarnoff Symposium*, Princeton, NJ, USA, 27-28 2006.

[26] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3776 – 3784, nov. 2005.

[27] T. Ohira, "Secret key generation exploiting antenna beam steering and wave propagation reciprocity," in *European Microwave Conference*, vol. 1, 4-6 2005.

[28] H. Koorapaty, A. Hassan, and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Communications Letters, IEEE*, vol. 4, no. 2, pp. 52 –55, feb 2000.

[29] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *IEEE ICASSP*, Las Vegas, NV, USA, March 2008.

[30] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 2, pp. 240 –254, june 2010.